REMARKS

1. Specification. Corrections are herein above submitted to the specification as requested in the Office action.

2. Claim Rejections. Claims 1-19 set out above are pending in the application and stand rejected under 35 USC 103(a) as being unpatentable over U.S. Patent No. 6,490,677 ("Aguilar") in view of U.S. Patent No. 6,393,458 ("Gigliotti") and further in view of U.S. Patent No. 6,601,096 ("Lassiter"). Applicant respectfully disagrees.

In addition, Applicant herein amends independent claims 1, 8 and 14 to incorporate additional limitations to even more clearly distinguish the present invention. Specifically, independent 1 and 8 claims are amended herein to state that the sever allocation table is updated to increment a particular boot server load count whenever that boot server sends an acknowledge to a requesting client. (Claim 7, as originally submitted, already included this limitation.) Claim 14 is herein amended to included limitations similar to amended claims 1 and 8. Claims 2, 9, 15 and 18 are herein canceled.

3. Claims 1, 7, 8 and 14. With respect to claims 1, 7, 8 and 14, the Office action relies first upon Aguilar, asserting that Aguilar teaches about clients, boot servers and a DHCP/PXE server. Aguilar does teach, among other things, about a pre-execution environment, in which a client is initialized from a boot server using a DHCP/PXE server, as in the present application.[1] However, it should be noted that Aguilar does not teach about allocating boot servers to clients by prioritizing the boot servers according to their loads in supplying boot images, as claimed in the present invention. This is because Aguilar concerns issues that arise due to multiple bootstrap programs residing on a client, which exist because the client may run a variety of operating systems, and does not concern prioritizing boot servers with regard to their loads in supplying boot images.[2]

---

[1] Aguilar, col. 5, lines 2-8 ("When a network computer 50a-50n is initially placed on network link 54, the user may be prompted to select a desired operating system or default to the first available operating system. The newly placed network computer, through a sequence of automatic operations, searches the network for an available server containing the desired operating system or default.").

[2] Aguilar, col. 2, lines 14-27 ("While factory loading of multiple bootstrap programs onto network computers provides for diversity of operating systems, configuration of the boot process of each network computer when initiated onto a network is required to direct the bootstrap program to the correct operating system kernel on a server and to set the correct bootstrap program for loading the operating system. However, it would be desirable for a network computer, upon initiation to a network, to automatically configure the network computer's boot process from the operating systems available on the network. Further, it would be desirable in cases where the user prefers a particular operating system to reduce the user steps necessary to configure a network computer for that

The Office action further relies upon Lassiter, asserting that Lassiter teaches about maintaining a Client Allocation Table associating client IP addresses with corresponding boot server IP addresses and providing the boot server IP addresses in the sequence of their listing in a Server Allocation Table for access whenever a client requests a DHCP. Specifically, the Office action states that Lassiter teaches about a client giving a DHCP server information about itself and the server giving server IP addresses.

With regard to this assertion, Applicant first notes that merely teaching about a client giving information about itself to a DHCP server and the DHCP server identifying server IP addresses does not teach or suggest maintaining a Client Allocation Table that associates client IP addresses with corresponding boot server IP addresses, and it especially does not suggest providing the boot server IP addresses in a certain sequence according to their listing in a Server Allocation Table for access whenever a client requests a DHCP, as in the present claimed invention. This is even more particularly true since the sequence of boot server listings in the Server Allocation Table is prioritized according to the boot server loads in supplying boot images.

Secondly, the cited passages of Lassiter do not even teach specifics about a DHCP server identifying server IP addresses. In one cited passage, Lassiter teaches about clients getting their *own* IP addresses from a DHCP server.[3] In another cited passage, Lassiter teaches about "discovering" a boot server, but specific details of this are not taught.[4] This is at least partly because Lassiter does not concern allocating boot servers to clients by prioritizing the boot

system.").

[3] Lassiter, col. 1, lines 25-41 ("As will be appreciated by those skilled in the art, protocol standards for Internet client server networks have been developed and adopted, including protocols for booting a client from a server. Computer networks that use the Internet protocol are commonly referred to as "IP networks." Within IP networks, host systems and other objects are identified as Internet Protocol Addresses (IP addresses). IP addresses provide a simple mechanism for identifying the source and destination of messages sent within IP networks. Increasingly, IP addresses within IP networks are assigned using the dynamic Host configuration Protocol (DHCP) defined in Internet RFC 1541 which is incorporated herein by reference. In networks that use the DHCP protocol, client systems request IP addresses from a DHCP server. The DHCP server allocates an IP address for use by the requesting client system and sends the client a message telling the client system which IP address to use.").

[4] Lassiter, col. 2, lines 11-27 ("In brief, the PXE protocol operates as follows. The client initiates the protocol by broadcasting a DHCPDISCOVER containing an extension that identifies the request as coming from a client that implements the PXE protocol. Assuming that a DHCP server or a Proxy DHCP server implementing this extended protocol is available, after several intermediate steps, the server sends the client a list of appropriate Boot Servers. The client then discovers a Boot Server of the type selected and receives the name of an executable file on the chosen Boot Server. The client uses TFTP to download the executable from the Boot server. Finally, the client initiates execution of the downloaded image. At this point, the client's state must meet certain requirements that provide a predictable execution environment for the image. Important aspects of this environment include the availability of certain areas of the client's main memory, and the availability of basic network I/O services.").

servers according to their loads in supplying boot images, as claimed in the present invention. Instead, Lassiter concerns a client giving a DHCP/PXE server information the server can use to determine more precisely the client machine's boot state in order to provide the client with a tailored boot image.[3]

With regard to the third reference relied upon in the rejection, Gigliotti, the Office action acknowledges that Gigliotti does not teach maintaining a Client Allocation Table associating client IP address with corresponding boot server IP addresses or providing IP addresses of boot servers in the sequence of their listing in a Server Allocation Table for access whenever a client requests a DHCP, as in the present claimed invention. However, the Office action nevertheless combines the averred teachings of Gigliotti with those of Aguilar and Lassiter. Specifically, the Office action contends that Gigliotti teaches about a DHCP/PXE server for allocating a boot server to each requesting client characterized in that the least loaded boot server is prioritized for service by maintaining a boot Server Allocation Table (SAT) containing the existing client load count for each boot server. The Office action also contends that Gigliotti teaches prioritizing the boot servers by sorting said SAT in order of increasing load count whenever it is updated.

As a threshold matter, Applicant notes that the Office action offers only a cursory contention as to what motivation could exist to combine the teachings of Gigliotti with those of Aguilar and Lassiter. This is particularly noteworthy, since Gigliotti does not even specifically teach about allocating boot servers to clients in a pre-execution environment according to the loads of the boot server with regard to supplying boot images, the subject of the claimed invention. Gigliotti concerns load balancing with respect to executables running on various

---

[3] Lassiter, col. 2, lines 37-62 ("Briefly, this invention contemplates the use of PXE Frame extension tags for remote boot loading client machines will afford servers deterministic ability for what image or utility the client requires based on its boot state. The invention takes advantage of the PXE frame by using the DHCP/PXE Vendor Tags for providing information to the PXE Server as to what image or boot process is required from the server by the client. This solution is targeted primarily, but not limited to, "media less" or "thin clients". The invention provides the client an ability to give a DHCP/PXE server more deterministic information about itself. The server can use this information (i.e. information contained within the Extension tags) to determine more precisely the client machine's boot state DHCP/PXE server code parses the DHCP/PXE extension tags (contained within the DHCP/PXE data frame) and uses the POST error information and/or the Vital Product Data (VPD) to provide the client with a tailored boot image. For example, this tailored image can supply a fix for the client POST error condition that was reported to the Server via the tag containing the POST error information. Conversely, the server can evaluate the VPD information contained within the tag to determine, for example, that the client POST code (firmware) is in need of updating and the server sends the appropriate boot image to the client. Also, the tag information (POST error and VPD) could be used in combination to allow the server to determine even a more appropriate boot image for the requesting client.").

machines.[6] More specifically, Gigliotti concerns load balancing of "hosts," which "generally refers to an actual computer that may be executing a plurality of virtual machines, processes or threads."[7]

Given that Gigliotti does not teach about DHCP servers for allocating boot servers to requesting clients in a pre-execution environment, Gigliotti does not even merely suggest a DHCP/PXE server for allocating a boot server to each requesting client, where the least loaded boot server is prioritized for service by maintaining a boot Server Allocation Table containing the existing client load count for each boot server. Even more certainly, Gigliotti does not suggest prioritizing the boot servers by sorting the Server Allocation Table in order of increasing load count whenever it is updated. Once again, this is particularly true given that the claims state the sequence of boot server listings in the Server Allocation Table are prioritized according to the boot server loads in supplying boot images.

The passages of Gigliotti that are relied upon in the Office action actually teach about load balancing among various "host" servers, i.e., servers executing a plurality of virtual machines, processes or threads for clients, and, in particular, teach about a load balancer monitoring and obtaining load readings at predetermined intervals of time for the hosts in order to determine a balanced distribution for the handling of "events" in the system.[8]

---

[6] Gigliotti, col. 2, lines 55-65 ("The invention provides a method, system and computer program product for load balancing in a distributed computing environment. The system for balancing the distribution of event messages in a distributed object computing environment includes at least one client publishing an event containing information and a plurality of server classes residing on one or more server hosts, at least one server class subscribing to the event published by the client, and a plurality of load balancers. Each load balancer queries the server hosts to calculate a load parameter for each server host.").

[7] Gigliotti, col. 1, lines 27-44 ("A simplified typical distributed object computing environment 10 is illustrated in FIG. 1. The distributed object computing environment 10 includes a plurality of machines 12. Machines 12 may be actual computers, such as work stations or PCs commonly known in the art or any other computers useful as either server or client machines, special purpose machines, or virtual machines. As used herein, the term "host" generally refers to an actual computer that may be executing a plurality of virtual machines, processes or threads. In the context of a manufacturing execution system, special purpose machines might include bar code software devices that operate on a computer, but appear to software that runs on and interacts with the virtual machine to be a complete computer. A common example of a virtual machine known in the art is the Java virtual machine, however, other types of virtual machines are available and may be used herein.").

[8] Gigliotti, col. 6, lines 37-57 ("In an exemplary embodiment, a load balancer object determines a balanced distribution for events which have been published or initiated by the client. In determining a balanced distribution for the handling of events in the system, the load balancer monitors and obtains a load reading for each host. The load reading is essentially a measurement of how "busy" the host is at the time the reading is taken. Monitoring of the load in the system can be performed intermittently at predetermined intervals of time. For example, in an exemplary embodiment, load monitoring or querying can be performed at intervals of 30 seconds. Clearly, other monitoring intervals are possible in accordance with the requirements of the system. For example, as the number of load balancing objects in a system increases, there is a tendency for a natural randomized distribution of load queries to develop. One load balancer may check for load on the servers every 5 seconds, while another may check every 15

Applicant notes that Gigliotti does teach that the load balancer builds a table of subscribing classes and associated server hosts, that the table uses the load on the server hosts and that the load balancer table ranks the server hosts based on their loads such that the load balancer can direct events to the highest ranking server host having a subscribing class to handle particular events.[9] However, nowhere is there a suggestion by Gigliotti of a pre-execution environment or that one of the "events" in the system is the supplying a boot image from a boot server to a client in a pre-execution environment. Moreover, even if it were to be assumed, merely for the sake of argument, that some substantial motivation were supplied for the combination of the references and that the teachings of Gigliotti were stretched so as to apply to supplying a boot image from a boot server to a client in a pre-execution environment, and to apply to ranking server hosts based on their boot-supplying loads, Gigliotti teaches that load readings are obtained at predetermined intervals of time,[10] which does not suggest prioritizing the boot servers in a list kept in a server allocation table by sorting the table in order of increasing load count whenever the table is updated, wherein the table is updated to increment a particular boot server load count whenever that boot server sends an acknowledge to a requesting client, as in the amended independent claims of the present application.

The Office action contends that Gigliotti teaches that updating a server allocation table, as claimed, is done to increment a particular boot server load count whenever that boot server sends an acknowledgment to a requesting client, i.e., contending that Gigliotti states there is an update of this sort "whenever the server has a thread to be sent to a client." Applicant respectfully disagrees. In the first place, there are substantial differences mentioned previously herein above with regard to boot servers versus host severs, pre-execution environment, etc. Secondly, it is not clear why it would be material to patentability in the present case if Gigliotti taught some sort of server load update whenever a host server had a thread to be sent to a client, since this is not what is claimed in the present case. Thirdly, even setting these issues aside, Applicant is unable

---

second intervals while the next at 30 second intervals and so on so that various overlapping patterns of load queries are taken at any given point in time.").

[9] Gigliotti, col. 6, line 57 - col. 7, line 24 ("Furthermore, in operation the load balancer builds a table of the subscribing classes and associated server hosts. The table is constructed using the list of subscribing classes for an event, and the load on the server hosts. . . . Finally, the load balancer table ranks the server hosts based on the load such that the server host with the lowest load will run all subscribing objects for the event. . . . The Load Balancer can then direct the event to the highest ranking server host having a subscribing class to handle the event, resulting in a balanced distribution of events among the server hosts.").

[10] Gigliotti, col. 6, lines 37-57.

to find any statement or suggestion by Gigliotti in the cited passage to the effect that a server load is updated whenever a host server has a thread to be sent to a client. Gigliotti *does* simply state that a "load value represents the number of system threads in queue waiting on the server host . . ." Col. 7, lines 28-31. As previously pointed out, Gigliotti teaches that this load value is updated at predetermined intervals of time. Gigliotti, col. 6, lines 37-57. This is different than updating whenever a host server has a thread to be sent to a client.

4. Claims 3-6, 10-13 and 16, 17 and 19. Claims 3-6, 10-13 and 16, 17 and 19 are patentability distinct merely due to their dependence upon amended independent claims 1, 8 and 14, respectively. Additionally, these claims include significant, nonobvious limitations that provide further basis for their allowability. However, in rejecting these claims the Office action essentially repeats the same language cited in the rejections of claims 1 and 2, 8 and 9, and 14 and 15. But claims 3, 10 and 16 state that the CAT is updated to include an entry associating the client with a particular boot server IP address whenever a boot server sends an acknowledge ACK to the client. The cited references do not teach or suggest this.

Likewise, in rejecting claims 4, 11 and 17 the Office action essentially repeats the same language cited in the rejections of claims 1 and 2, 8 and 9, and 14 and 15. But claims 4, 11 and 17 state that the CAT is updated to remove an entry corresponding to a particular client whenever the DHCP refreshes it's IP addresses pool and discovers that said client is not available. The cited references do not teach or suggest this.

Likewise, in rejecting claims 5, 12 and 19 the Office action essentially repeats the same language cited in the rejections of claims 1 and 2, 8 and 9, and 14 and 15. But claims 5, 12 and 19 state that the SAT is updated to decrement the load count on a particular boot server using the association between the client and server given in the CAT whenever the DHCP refreshes it's IP addresses pool and discovers that said client is not available. The cited references do not teach or suggest this.

Application No. 09/655,093 Atty. Docket JP920000177US1
Filing Date 09/05/2000 Reply A to Office action of 10/23/2003

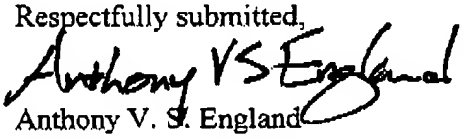
PRIOR ART OF RECORD

Applicant has reviewed the prior art of record cited by but not relied upon by Examiner, and asserts that the invention is patentably distinct.

REQUESTED ACTION

Applicant contends that the invention as originally claimed is patentably distinct, and hereby requests that Examiner grant allowance and prompt passage of the application to issuance.

Respectfully submitted,



Anthony V. S. England
Attorney for Applicants
Registration No. 35,129
512-477-7165
a@aengland.com